Chemical Engineering Science 192 (2018) 678–687

Contents lists available at ScienceDirect

Chemical Engineering Science

journal homepage: www.elsevier.com/locate/ces

A novel metaheuristic framework for the scheduling of multipurpose batch plants

Matthew Woolway [a,b], Thokozani Majozi [b,*]

[a] School of Computer Science and Applied Mathematics, University of the Witwatersrand, Johannesburg, Private Bag 3, Wits 2050, South Africa
[b] School of Chemical and Metallurgical Engineering, University of the Witwatersrand, Johannesburg, Private Bag 3, Wits 2050, South Africa

HIGHLIGHTS

- An evolutionary framework for scheduling of multipurpose batch plants.
- A method that does not rely on time or event points.
- No need for binary variables.
- The method does not suffer scalability issues of mathematical programming methods.
- Solution times reduced by almost 98% in sufficiently long time horizons.

ARTICLE INFO

Article history:
Received 8 October 2017
Received in revised form 25 June 2018
Accepted 14 August 2018
Available online 16 August 2018

Keywords:
Scheduling
Metaheuristics
Batch plant
Multi-purpose
Optimisation

ABSTRACT

A genetic algorithm (GA) is proposed along with a general framework for the scheduling of a typical multipurpose/product batch plant. The majority of literature regarding these problems make use of mathematical programming methods. Modelling problems in this manner leads to numerous binary variables relating to material balance and sequence of batches along long time horizons, thus resulting in large computational time. The proposed GA does not suffer the same scalability issues of mathematical programming approaches. The GA makes use of a coupled chromosome system with specific crossover and mutation functions utilised with the purpose of profit maximisation. Results show that optimal or close-to-optimal solutions can be achieved with a reduction of up to 98.53% computational time in certain cases.

1. Introduction

Over the past two decades batch process research has arisen in both industrial and academic domains alike. Economic changes have led to a demand for high value – small volume products. The flexible nature of batch plants allow for the production of multiple products utilising the same facility. However, this benefit results in increased complexity in the scheduling of the facility. Plant configurations differ in complexity, ranging from simpler single-stage plants (numerous parallel machines) to challenging multistage and multipurpose plants. A large and growing amount of research focusses on the development of optimisation techniques to determine these schedules, specifically schedules that minimize the time required to attain a given objective value. Ideally these techniques should find this objective value in the minimum computational time possible.

Earliest work in process scheduling primarily focussed on flowshop and general multistage batch plants while seminal work on multipurpose batch plants was proposed by Kondili et al. (1993). Unlike the complexity of multistage plants, solving for the schedules of multipurpose plants is inherently NP-Hard due to the vast combinatorial enumeration. The majority of mathematical formulations pertaining to multipurpose batch plant scheduling are based upon three major representations, namely: (i) the state task network (STN) introduced by Kondili et al. (1993); (ii) the resource task network (RTN) proposed by Pantelides (1994); (iii) the state sequence network (SSN) developed by Majozi and Zhu (2001). These three representations can be applied to any situation ranging from batch to continuous processes which may consist of tasks both variable and constant in nature. Models utilising these formulations are able to account for various complex network situations such as recycle streams, mixing/splitting batches, variable batch

* Corresponding author.
E-mail address: thokozani.majozi@wits.ac.za (T. Majozi).

https://doi.org/10.1016/j.ces.2018.08.031
0009-2509/

Nomenclature

α	rate for neighbourhood radius
$\bar{v}^i$	vector of volume states at given event point i
β	mutation rate for child mutation
γ	selection rate for parent selection
c	integer crossover point for chromosomes
H	the time horizon
$I = \{i \mid i = \text{event point}\}$	the set of all event points
$J = \{j \mid j = \text{unit}\}$	the set of all units
$n = \mid I \mid$	the number of event points in the set I
$subs_j^i$	instruction set for unit j at event point i

sizes, while also satisfying storage scenarios such as; no intermediate storage (NIS), finite intermediate storage (FIS) and unlimited intermediate storage (UIS). These models generally invoke the optimisation of the maximisation of profit over a known fixed time horizon or minimisation of makespan, where the number of batches is known a priori.

In any of the aforementioned formulations, time can be handled with either discrete-time models or continuous-time models. Pioneering models initially made use of discrete-time representations (Kondili et al., 1993). The major drawback in these cases was that in order to discretise the time horizon into a meaningfully accurate way, the number of binary variables scaled into the several thousands. The tractability of this approach drops off rapidly as soon as one attempts to solve any large size problems. Subsequent work on discrete-time representations was done by Shah et al. (1993). Here, task process times are set to be constant, with the time horizon carved into intervals of the greatest common factor of all task processes. Flaws exist when considering constant processing times as they often may not be plausible due to task process times potentially differing significantly. A further concern is interval widths which could become unnecessarily small leading to an exponential increase in the number of intervals required, rendering the model unsolvable. This gave rise to the development of the continuous-time representation (utilising all three network representations), notably by Schilling and Pantelides (1996), Zhang and Sargent (1996), Ierapetritou and Floudas (1998), Mockus and Reklaitis (1999), Majozi and Zhu (2001), Giannelos and Georgiadis (2002) and Maravelias and Grossmann (2003). Variable process times were now possible and the time horizon was segmented into intervals of unknown and unequal duration. This approach allowed for more realistic scenarios to be modelled and solved. These intervals are achieved through the use of event points. Specifically, the event points note the position on the time domain where a task(s) begins or ends. A detailed review is not planned here but the interested reader is referred to the excellent reviews done by Floudas and Lin (2004), Méndez et al. (2006) and Shaik et al. (2006). A major drawback in this approach is that the number of event points required to find a global optimum is not known beforehand. Common practice appears to slowly increase this number until the objective value no longer improves. While this may be true in some cases it simply cannot be stated that this is generally true. Interestingly, this suggests that the claim for global optimality should not necessarily be claimed but rather hypothesised and stated as the best optimum *thus* far.

The above advances in continuous-time formulations and STN/SSN approaches primarily attempt to reduce the number of binary variables in the formulation, thus allowing for a more feasible computational solution time. When considering larger scale scheduling problems however, the number of binary variables is set to scale regardless of the quality of the representation, resulting in intractable solution times. Since the aforementioned issue regarding the selected number of event points required exists, the suggestion of non-deterministic techniques being applied to these problems seems relevant. This is especially true when considering the scalability of these techniques when large problem sizes are involved. Some work has been conducted in the application of metaheuristic techniques to batch processes, such as a genetic algorithm (GA) in Azzaro-Pantel et al. (1998) showing promise for large scale combinatorial problems. Cantón Padilla (2003) illustrates speed comparisons between mixed integer linear programmes (MILP) and a GA in the scheduling of chemical batch plants. The majority of this early work either focussed on the design of batch plants using metaheuristics such as Cavin et al. (2004) or the single-stage or multiproduct batch plant scheduling as found in He and Hui (2006, 2007). In terms of solving multipurpose process scheduling through metaheuristics, very little literature exists. He and Hui (2010) introduce an extremely fast GA which comprehensively out-competes top performing MILP models by orders of magnitude. Unfortunately, the framework in which the GA is designed exploits the nature of the case study solved as well as considering constant process time. Although this is a useful contribution, it lacks a generalised framework to apply to other problems.

This work investigates a well-posed and well understood case study of a multipurpose batch plant. Originally presented in Kondili et al. (1993), the example contains batch mixing and splitting as well as a recycling stream. In our approach we do not make use of the STN in the holistic sense but obey the constraints and material balances outlined by it. In addition, the flowsheet is also involved and the state of this is monitored through the solution technique. The solution technique used here is a generalised GA involving two chromosomes. The remainder of the paper is layed-out in the following way. Section 2 outlines the problem description, Section 3 illustrates the solution mechanism, Section 4 discusses results for both a motivating example as well as the illustrative case study and Section 5 pertains to the overall discussion and conclusions.

2. Problem description

Before introducing the new framework and solution method, we discuss the two examples to be utilised. In Figs. 1 and 2, we have a simple multiproduct literature example introduced by Ierapetritou and Floudas (1998). Table 1 provides the pertinent data. Here we have a single product produced in the multiproduct facility, involving mixing, reaction and purification stages. The specific literature example has been well utilised when introducing a new framework. The primary literature example used as introduced by Kondili et al. (1993) is described in Figs. 8 and 9 respectively.

In both examples we consider variable processing times. As mentioned previously, most approaches thus far have made use of mathematical programming. These have been successful in handling constraints when dealing with assigning of batch tasks to units while preserving the material balances for the resultant states. They also effectively handle the syntactical structure of the plant layout such as sequence dependence and storage